

INFORMATION RETRIEVAL MEMORY BY PREFIX ANALYSIS FOR BUILDING-UP ROUTING TABLE OF NODE FOR HIGH SPEED COMMUNICATION NETWORK SUCH AS INTERNET

Publication number: JP2000151691 (A)

Publication date: 2000-05-30

Inventor(s): FILIPPI ENRICA; INNOCENTI VIVIANA +

Applicant(s): CSELT CENTRO STUDI LAB TELECOM +

Classification:

- **international:** **H04L12/56; H04L12/56;** (IPC1-7): H04L12/56

- **European:** H04L12/56C

Application number: JP19990304159 19991026

Priority number(s): IT1998TO00909 19981027

Also published as:

JP3299528 (B2)

EP0998162 (A2)

EP0998162 (A3)

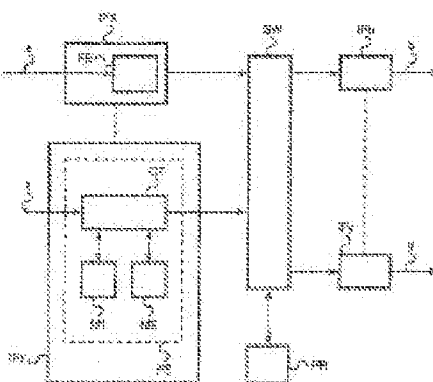
EP0998162 (B1)

US6571313 (B1)

[more >>](#)

Abstract of JP 2000151691 (A)

PROBLEM TO BE SOLVED: To provide a memory structure that ensures incremental update of a routing table and optimum utilizing of a physical memory. **SOLUTION:** A memory to retrieve information by a prefix analysis in order to build up a routing table for a node of a high speed communication network such as the Internet especially includes a memory element M1 that stores one set of information items and each of the information items relates to mask information and target information denoting a number of important characters in respective prefixes. In order to realize a retrieval reference based on coincidence of a longest prefix, each cell includes an information field to provide an address of a succeeding line for consecutive retrieval or an arrived target and a couple of flags specifying contents of the information field.; An auxiliary vector AUX includes cells of the same number as memory lines and when a flag in a cell of the memory element indicates necessity for retrieval of a succeeding line and read of the target, the target information is stored in a cell relating to the succeeding line.



.....
Data supplied from the **espacenet** database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-151691
(P2000-151691A)

(43) 公開日 平成12年5月30日 (2000.5.30)

(51) Int.Cl.⁷

H 0 4 L 12/56

識別記号

F I

H 0 4 L 11/20

テーマコード* (参考)

1 0 2 D

審査請求 有 請求項の数 9 O L (全 21 頁)

(21) 出願番号 特願平11-304159

(22) 出願日 平成11年10月26日 (1999. 10. 26)

(31) 優先権主張番号 T O 9 8 A 0 0 0 9 0 9

(32) 優先日 平成10年10月27日 (1998. 10. 27)

(33) 優先権主張国 イタリア (I T)

(71) 出願人 591063103

クセルトーセントロ・ステュディ・エ・ラ
ボラトリ・テレコムニカチオーニ・エツ
セ・ピー・アー

CSELT-CENTRO STUDI
E LABORATORI TELECO
MUNICAZIONI SOCIETA
PER AZIONI

イタリア国 トリノ、10148 ヴィア・グ
リエルモ・レイス・ロモリ 274

(74) 代理人 100064355

弁理士 川原田 一穂

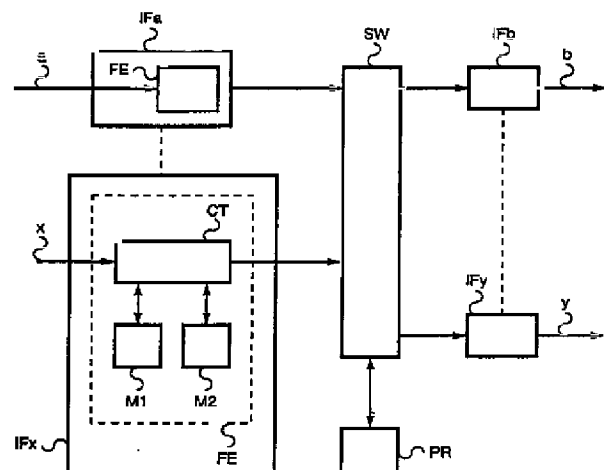
最終頁に続く

(54) 【発明の名称】 インターネット網等の高速通信網のノードのルーティング表構築のためのプレフィックス分析による情報探索用メモリ

(57) 【要約】

【課題】 物理メモリの最適利用とルーティング表のインクリメンタル更新を保証するメモリ構造を提供すること。

【解決手段】 特にインターネット網のような高速通信網のノードのルーティング表を構築するべくプレフィックス分析により情報を探索するためのメモリは、1組の情報項目を記憶するメモリ要素 (M1) を含み、該情報項目の各々は、それぞれのプレフィックスにおける重要文字数を示すマスク情報とターゲット情報に関連する。最長プレフィックス一致に基づく探索基準を実現するために、各々のセルは、連続探索のための次行のアドレス又は到達ターゲットに関する情報のどちらかを与える情報フィールドと、情報フィールドの内容を特定する一対のフラグ (GO, TARGET) とを含む。補助ベクトル (AUX) は、メモリ行と同数のセルを含む。該補助ベクトルは、メモリ要素のセル内のフラグが、次行の探索動作を行う必要と共にターゲットの読み出しを示すとき、ターゲット情報を前記次行に関連するそのセルに記憶する。



【特許請求の範囲】

【請求項1】 個々の情報項目の最重要部を構成するプレフィックスを分析することに基づいた情報探索を行うためのメモリであって、(ア)1組の前記情報項目を記憶する第1メモリ要素(M1)であって、各々の情報項目は、マスク情報とターゲット情報に関連し、マスク情報は、それぞれのプレフィックス内の重要文字数を示し、ターゲット情報は、場合によっては別の情報の組にアクセスするのに利用できるデータを構成し、前記情報項目は、行と列に編成されたそれぞれのメモリセルに記憶される上記第1メモリ要素(M1)と(イ)メモリ内の特定情報項目の探索を制御し、かつ、メモリを更新する制御装置(CT)であって、文字から成る入力ストリングにおける所定長の連続部分と、前記部分の長さの倍数ではない可変長を有する記憶されたプレフィックスの対応部分とを比較することで動作する上記制御装置(CT)を含み、最長プレフィックス一致に基づいた探索基準、すなわち結果として入力ストリングとの最長プレフィックス一致を与える探索基準を実施するために、

(1) 各々のセルが、情報フィールドと一対のフラグ(GO, TARGET)を含み、該情報フィールドは、探索を継続するための次行アドレスから到達ターゲットに関する情報のどちらかを与え、前記一対のフラグは、情報フィールドの内容を特定し、そして(2)メモリの行と同数のセルを有する補助ベクトル(AUX)が設けられ、該補助ベクトルは、メモリ要素(M1)内の前記セルのフラグが、次行の探索を継続する必要性を伴ってターゲットに到達したことを示すとき、ターゲット情報を前記次行に関連するセルに記憶するように構成され、前記補助ベクトルの各セルは、前記メモリ要素(M1)のセルのものと同一の情報フィールド及び一対のフラグを含む、ことを特徴とする上記メモリ。

【請求項2】 メモリ要素(M1)及び補助ベクトル(AUX)の前記セル内の情報フィールドは、到達ターゲットの情報用に、ターゲット識別及びそのターゲットに関連するマスクを表す値を記憶することを特徴とする請求項1記載のメモリ。

【請求項3】 到達ターゲットの情報が複数のセルに共通するとき、前記複数のセルのうちの一つにおける前記情報フィールドも、到達ターゲットに関連するプレフィックスによりカバーし得るより短いプレフィックスに関する補助情報を記憶することを特徴とする請求項2記載のメモリ。

【請求項4】 前記セルが、行内にて、調べられる入力ストリング部分、その部分に関連するマスク、及びその部分の長さにより一対一対応したアドレスに置かれることを特徴とする請求項3記載のメモリ。

【請求項5】 このようなメモリが、高速通信網のノードに対するルーティング表を実現し、前記プレフィックスが、前記通信網にて伝送される情報に関連する宛先ア

ドレス又は宛先アドレスの部分であることを特徴とする請求項1〜4のいずれか一項に記載のメモリ。

【請求項6】 前記通信網がインターネット網であることを特徴とする請求項5記載のメモリ。

【請求項7】 個々の情報項目の最重要部分を構成するプレフィックスの分析に基づいた情報探索用メモリを管理する方法であって、

前記情報は、それぞれのプレフィックス内の重要文字数を示すマスク情報、及び別の情報の組へのポインターとして使用されるターゲット情報と共に、メモリ要素(M1)のセルに記憶され、前記メモリ管理では、受信した文字ストリングの連続部分と、前記部分の長さの倍数ではない可変長を有する記憶されたプレフィックスの対応部分とを比較することを要し、最長プレフィックス一致に基づいた探索基準を実施するために、(1)メモリ要素(M1)の各セルは、一対のフラグ(GO, TARGET)に関連し、該一対のフラグは、セルが属する行にて探索操作が終了するか又は次の行に継続しなければならないかを、それらの論理値により特定し、後者の場合には、可能なプレフィックスとの一致が特定行において見出されたならば、ターゲットに到達しており、(2)メモリ要素(M1)の全ての行は、補助ベクトルのセルに関連し、該補助ベクトルは、探索が終了する行ではない行に対応して到達するターゲットに関する情報を記憶し、(3)前記部分の長さの倍数である長さを有するプレフィックスに対応するセルの組内にて選択されたセルもまた、より短いプレフィックスに関するターゲット情報を記憶し、該より短いプレフィックスは、前記部分の長さの倍数長を有さず、倍数長を有する前記プレフィックスによりカバーされ得、前記セルは、探索の最終セル又は探索中の中間セルのどちらであるかに従って、メモリ要素(M1)又は補助ベクトル(AUX)のセルである、ことを特徴とする上記方法。

【請求項8】 情報項目を削除するため、このような情報項目を含む行に到達するとき、(a)より短いプレフィックスに関連するターゲットにより表される可能な代用ターゲットに、メモリ又は補助ベクトル内のセルの前記組内にて到達し、(b)このような代用ターゲットが存在するならば、それを削除されるべきプレフィックスの位置に記憶する、ことを特徴とする請求項7記載の方法。

【請求項9】 削除される情報項目の探索中に調べられる行全ての識別が、一時的に記憶され、また、置換しないターゲット削除の場合、情報削除の結果空となった行内での後方チェックが行われ、関連識別が、メモリに関連する空行リストに記憶されることを特徴とする請求項8記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、高速通信網、好ま

しくはインターネット網のルーティング表を構築するのに特に専用されるプレフィックス分析による情報探索用メモリに関する。

【0002】

【従来の技術】ATM網、インターネット網などのような現在の高速網の通信プロトコルは、情報パケットの伝送に基づいており、該情報パケットは、パケットの宛先アドレスに関する情報をヘッダー内に含んでいる。網ノードでは、パケットを受信すると、ルーティング制御装置が、アドレスを用いて適当な表又はデータベース内部を探索し、出力インターフェースを捜し出す。パケットは、該出力インターフェースに送られ、さらに後続のネットワーク装置に発送される。現在のところ、Gbits/sのオーダーの伝送速度が普通であり、よって、ノードでのルーティング時間は、通常のトラフィック流を害することを防止するべく非常に短くなければならない。量的に示すために、約1000ビットのパケットの1Gbit/s伝送を考えると、ノードは、1秒当たり約100万個のパケットを扱うことを要求される。すなわち、スイッチング時間、場合により必要なパケットヘッダー処理時間、及び場合により必要な連続パケット間のガード時間を考慮すると、最も長い操作であるルーティング表の探索は、1 μ sよりやや小さい時間内で実行しなければならない。通信速度が増すにつれて、状況はますますクリティカルとなる。

【0003】記載を簡単にするため、単なる例としてインターネット網を見てみると、インターネット網を使用する新しいアプリケーションの絶え間ない発展にもより、帯域要求のみならずネットワークに接続されたホストコンピューターの数及びトラフィックの両方が指数関数的に成長している。このこと全てが、ルーティング表のサイズを増大させる結果となり、それにより、より長い探索操作を生じさせる。大きなメモリ容量を低コストで得るのはますます容易になっているが、大きなメモリへのアクセス時間を低減するのは相当の技術的な問題をはらんでいることは、技術者には周知である。ネットワークの拡張に容易に適応するために、インターネットアドレスは、階層構造により編成され得る。この意味で、例えば同じ地理領域又は同じプロバイダーに対応するアドレスグループは、最上位アドレスビット（プレフィックス）を共有する。これらのプレフィックスは、可変長を有し、他のプレフィックスの最上位部（プレフィックスのプレフィックス）を構成し得る。インターネットアドレスの構造のさらなる詳細は、例えばH.P.Giesigerによる文献「Das Internet-Protokoll der naechsten Generation」（Comtec 5、1998年、第20～26頁）、又はR.M.Hindenによる文献「IP Next Generation Overview」（1995年5月14日、インターネットサイト<http://playground.sun.com/pub/ipng/html/INET-Ipng-Paper.html>にて入手可能）が参照できる。

【0004】探索の複雑さを軽減するために、何よりもまず、ネットワークにおけるルーティング情報が、処理されているパケットの宛先に通じる経路内の次ノードのアドレスに関係していることが考慮される。そのアドレスは、普通は「次ホップ(next hop)」なる用語により示される。ルーティング表において可能な異なる次ホップは、百のオーダーであるから、関連情報を抽出して別のデータ構造に記憶することが可能である。小さな大きさの場合には、リテラチャー(literature)「TARGET T」で指定されたインデックスにより直接アクセスすることにより、このデータ構造にアクセスし得る。該リテラチャー「TARGET T」は、受信したパケットのアドレスから識別される。実際には、ルーティング管理装置において、パケット宛先アドレスを検索キーとして用いることにより、発送表へのアクセスが得られる。発送表は、ネットワークの完全なルーティング表の部分集合であり、各行は、プレフィックス（アドレスマスク対。ここで、マスクは、アドレスの上位ビット数、すなわちプレフィックスを構成するビット数を示す。）と対応するデータ（さらなる情報と共にターゲット、普通は、出力インターフェースへのポインター）を含む。アドレスに関連するマスクとキーのビット毎のANDが、アドレス自身に等しいならば、プレフィックスは探索キーに対する一致を生じる（すなわち、ターゲットに到達することを許容する）。

【0005】インデックス探索は複数のターゲットを探索する結果となるので、プレフィックスの長さの可変性は、さらなる複雑さをもたらす。というのは、比較により、異なる長さの幾つかのプレフィックスに対して正の結果が与えられるからである。これらの条件では、実際のターゲットは、最も長いプレフィックス、すなわち宛先に対して最も特異なプレフィックスに対応するものである。この基準は、「最長プレフィックス一致(longest prefix match)」として知られている。理論的には、最長プレフィックス一致に基づいた探索を行う最も効率的で最速の方法は、情報項目（エントリ）を1以上の3進CAM（内容アドレス指定可能メモリ(Content Addressable Memory)）に記憶することである。3進CAMでは、各々のメモリビットは、3つの値、すなわち0、1及び「ドントケア(don't care)」を取り得る。比較レジスタのビットと「ドントケア」論理値に設定されたメモリビットの比較は、明らかに正である。従って、データ自身の関連ビットをドントケア・レベルに単に設定することにより、データに関連したマスクを暗黙のうちに実現することが可能である。この種の装置の欠点は、主に高い値段と集積化能力の低さに存する。

【0006】低コストの従来メモリが使用できるためには、データベースにおける探索アルゴリズムであって速くてコンパクトなアルゴリズムを利用する必要がある。探索アルゴリズムの速度は、厳密にはメモリアクセス回

数に関係し、コンパクト性は、使用されているデータ構造を記憶するのに必要なメモリ量で決まる。最終的な目的は、探索プロセスで情報を記憶するのに要するアクセス数とメモリサイズの両方をできるだけ最小化することである。最長プレフィックス一致の基準を満足するのに適切なルーティングアルゴリズムは、実際にはソフトウェア型とハードウェア型の両方で実現され得る。いずれにしても、一般には、それらは以下のような所定数の共通要求を満たさなければならない。

(1) テーブルの大きさから出来るだけ独立した探索速度

(2) アルゴリズム性能における最悪の場合と最高の場合の間での小さな相違

(3) 表の容易なインクリメンタル(又はローカル)更新(すなわち、幾つかの情報項目の挿入又は取消は、表全体又は表の実質的な部分を再書き込みすることを要求しない。)

(4) 将来の技術発展やアドレス編成の変化に容易に適応するための柔軟性

(5) 浪費を防ぐため出来るだけ規則的なメモリ編成

【0007】最長プレフィックス一致を実施する必要性を考慮するような方法にてインターネット網ルーティングの問題を解決するために、幾つかの解決策が既に提案されている。第1群の解決策は、「PATRICIA」として公知のアルゴリズムに基づく。これは、大きなファイル、実際にはライブラリーのカタログにおける情報の記憶、インデックス付け、及び検索のためのアルゴリズムである。このアルゴリズムの原理は、D.R.Morrisonによる記事「PATRICI - Practical Algorithm To Retrieve Information Coded In Alphanumeric」(Journal of the Association for Computing Machinery, Vol.15, No.4, 1968年10月、第155頁以降)に記載されており、インターネット網のルーティング問題を解決することにおけるその応用は、関連の技術文献に広く文書化されている。PATRICIAアルゴリズムは、実質的にはアドレスビット上に構築されたプレフィックス木を操作するアルゴリズムである。ここで、各ビットはノードに対応する。アドレスビットは、一度に一つずつ調べられる。存在しないブランチに対応するノードを調べるのを避けるために、各ビットは、調べられる次ビットの指標に関連する。探索が終了すると、プレフィックスの現実の存在がチェックされる。各ノードは、以上のメモリアクセスに対応するので、メモリアクセスがデータ処理より十分に遅いことを考慮すると、本アルゴリズムは、アドレスを検索するための合計時間を事実上低減する。

【0008】PATRICIAアルゴリズムに基づいた解決策の本質的な利点は、実際には最悪の場合に対して校正が実施されていることであり(実際、全ての木ノードの通過が考慮されなければならない)、一方、現在の

インターネット網では、統計的に約20個のノードが、ターゲット(探索されるアドレス)に到達する前に分析されなければならないことが分かっている。従って、これらの解決策は、本質的に非常に遅く、加えて、最悪の場合と典型的な場合の間に大きな相違を示す。探索プロセスを加速するために、幾つかの解決策が提案されている。これらの解決策では、1ビットのみの代わりに複数のアドレスビットが、木に沿った各検索ステップにおいて考慮される。T.Pei とC.Zukowskiによる記事「Putting Routing Tables in Silicon」(IEEE Network Magazine, 1992年1月、第42頁以降)は、これらのアルゴリズムが所謂「トリ・メモリ(tric memories)」によりハードウェア内でいかにして実現され得るかを開示する。この場合の主要な問題は、複数の一致が存在しない場合にのみ従来のトリ・メモリが利用できることである。従って、ルーティング表に必要なメモリを低減し又は一般に探索プロセスを固定するような方法にて、インターネット網のルーティング表を探索するアルゴリズムの多くの提案は、プレフィックスを再構成して複数一致を除去することに基づいている。この様にしてこれらのアルゴリズムは、理論的にはトリ・メモリを使用して実現し得る。

【0009】プレフィックスを再構成する例が、M.Degemmark他による論文「Small Forwarding Tables for Fast Routing Lookups」(コンファレンスACM SIGCOMM '97, Cannes, 仏国、1997年9月16~18日)、及びV.SrinivasanとG.Vargheseによる論文「Faster IP Lookups using Controlled Prefix Expansion」(コンファレンスACM SIGMETRICS '98, Madison, ウィスコンシン州、米国、1998年6月22~26日)に記載されている。Degemmark他により提案された解決策は、3レベル木を示し、その各々のレベルは、アドレスの16、8及び8ビットをそれぞれカバーし、場合によっては適当なプレフィックスの再構成により作られた特にコンパクトな構造内に記憶される。それにより、木における各々のプレフィックスは、アドレスのインターバルを捜し出す。この解決策は、PATRICIAに基づく解決策(12アクセスを予測する最悪の場合約400ns)に対し、要求されるメモリ量と探索のためにメモリ自身にアクセスする回数を劇的に最小化する。しかし、プレフィックスを再構成することにより、インクリメンタル更新が不可能となり、各更新にてメモリを完全に再書き込みすることが要求される。更新プロセスは、非常に遅く、正常なネットワーク管理に対して問題を生じさせる。

【0010】V.SrinivasanとG.Vargheseにより提案された解決策は、プレフィックスが限定数のみの長さ値(場合によっては可変)をとるように、プレフィックスを再構成し、かつ、最短のプレフィックスを所定の最も近接した長さ値を有するプレフィックスの組と置き換えるこ

とにより、最短のプレフィックスを拡張する。ルーティング表は、特定数の副表に変換され、該副表の各々がポインターを次の副表に記憶する。このような副表の長さは、所与のアクセス数に対して、プレフィックスのその特定の組のために必要なメモリを削減するように最適化される。この構成により、たとえば表の長さが一般には予測できなくて物理メモリの最適利用が得られないとしても、使用されるメモリ量の削減が、一般に得られる。加えて、結果として得られるデータ構造は、プレフィックスの特定の組に対して最適化されるので、それはほとんど一般的でない。

【0011】

【発明が解決しようとする課題】本発明は、物理メモリの最適利用とルーティング表（又は一般にエントリの組）のインクリメンタル更新を保証する構造を提供することを目的とする。

【0012】

【課題を解決するための手段】本発明によるメモリは、（１）複数行及び列のメモリセルを含むメモリ要素であって、各セルは、１組の情報項目に属する１項目を記憶し、各情報項目は、それぞれのプレフィックスの幾つかの重要文字を示すマスクに、及び場合によっては別の組の情報項目をアクセスするのに利用できるデータを構成するターゲットに関連する前記メモリ要素、及び（２）メモリ内の特定の情報項目を探索するのを制御し、かつ、メモリを更新するための制御装置であって、該制御装置は、入力にて受信した文字ストリングの所定長の連続部分と、記憶されたプレフィックスの対応部分を比較することにより動作し、前記プレフィックスは、場合によっては前記部分の長さの倍数でない可変長である前記制御装置を含み、最長プレフィックス一致に基づいた探索基準を実施するために、（ア）各セルは、情報フィールド及び情報フィールドの内容を特定する一対のフラグに細分され、情報フィールドは、探索を継続するための次行アドレスに到達ターゲットに関する情報のどちらかを記憶し、そして（イ）補助ベクトルが与えられ、該補助ベクトルは、前記メモリ要素内の行と同数のセルを有し、また、該補助ベクトルは、メモリ要素の前記セルの一つのフラグが、次の行への探索の遂行の必要性和共にターゲットの到達を示すとき、ターゲット情報を前記次の行に関連するセルに記憶でき、前記補助ベクトルの各セルは、前記メモリ要素のセルのタスクと同じタスクを有する一対のフラグと情報フィールドを記憶することを特徴とする。本発明は、上記記載のメモリの管理方法にも関する。

【0013】

【実施例】詳細な説明のため、添付図面が参照される。図１は、インターネット網へのルーティングノードの基本図である。図２は、本発明によるメモリの編成を示す図である。図３Ａ～３Ｃは、メモリセル内のデータ編成

を表す。図４及び５は、図２のメモリ内での探索操作に関する図である。図６は、探索の数値例である。図７～１５は、本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【0014】図１では、インターネット網のようなネットワークのノード内のルーターが、１組の入力インターフェース IFa, \dots, IFx と出力インターフェース IFb, \dots, IFy により略示されている。インターフェースの各々は、一方の側では、 a, x, b, y で一般的に示されたラインに接続され、もう一方の側では、スイッチング装置 SW に接続されている。このスイッチング装置 SW は、制御装置又はネットワークプロセッサ PR に接続される。一般に、入力インターフェースは、入来データ用のバッファ（図示せず）と、情報処理装置を含む。情報処理装置は、ノード内のルーティング探索を管理し、すなわち、入力インターフェースに入来したパケットが発送されるべき出力インターフェースを捜し出す。プロセッサ PR は、ノードの通信及び制御の管理に要求される他の全ての処理を行う。異なるルーター構成では、ルーティング探索は、特定の処理装置に割り当てることができる。ルーティング探索装置は、全体として FE で示される。

【0015】この探索を実行するため、装置 FE は、ノード自身から到達可能な宛先のプレフィックスに関するルーティング表の一部の局部複製を含むべきである。冒頭記載のように、ルーティング表は、２つのレベル上にて構成され得る（従って、２つの記憶装置を利用する）。すなわち、第１のレベルでは、パケットアドレスを用いて、インデックス又はターゲットが、その宛先アドレスの表に含まれる最長プレフィックスに対応して捜し出され、一方、第２のレベルでは、このようなインデックスにより、パケットを発送するのに要求される情報へのアクセスが、直接得られる。これら２つの装置は、インターフェース IFx に対して表示されており、それぞれ $M1, M2$ で示される。参照符号 CT は、処理制御装置を示し、表内での探索及び表の更新を可能にする。本発明は、最長プレフィックス一致に基づいた探索を実施するために拡張された所謂マルチビット・トリ・メモリの形式のメモリ $M1$ を実現することに関する。

【0016】図２では、例えば各探索ステップにて最大 k ビットのプレフィックスが考慮されていると仮定して、本発明によるメモリの論理構造が示されている。図２から分かるように、トリ・メモリは、 $L=2^k$ 列のマトリックスにより従来様式にて作られる。各々の列は、一度に k のアドレスビットと R 個の行の可能な組み合わせのうちの一つに対応する。これは、アドレスの数及び分布、並びに k の値に依存する。各々のメモリセルは、空とすることができ（すなわち、記憶されたどのアドレスにも存在しない組み合わせに対応する）、又は到達されるべき次行のアドレス、及び／又はターゲットが到達

されたことに関する指標を含み得る。探索が最長プレフィックス一致に基づいている場合、2つの指標「到達ターゲット(target reached)」と「次行へ通過(passage to the successive row)」は、相互に排他的ではなく、これは、データを編成するとき、考慮されねばならない。この目的のため、セルが次行のアドレス用のフィールドと「到達ターゲット」指標用のフィールドを含む構造を採用することが明らかに可能である。しかし、この解決策は、メモリの無益な浪費をもたらす。というのは、2つの指標が同時に存在することは、例外であり、一般的でなく、その結果、ほとんど全てのセルに対し、フィールドの一つは空となるからである。また、文献に記載された解決策は、多重一致を除去し且つターゲットを木の最終レベルにシフトするようにプレフィックスを再構成又は拡張することを考慮しているが、好都合でない。というのは、このような解決策は、特に表内のエントリ数が多い場合により大きなメモリの大きさを要求し、かつ、インクリメンタル更新を困難なものとするからである。

【0017】上記記載の要求を満足するために、本発明によるメモリの各セルは、単なる情報フィールド、及び一対のフラグ(GOとTARGET)により形成される。これらのフラグの組み合わせが、情報フィールドの内容の意味を定める。情報フィールドは、(32ビット上のインターネットアドレスに関する現在のケースでは)例えば22ビットを含み得、その結果、各セルは、全体で3バイトから成る。第1フラグGOは、探索を継続すべきか否かを決定することを可能にし、第2フラグTARGETは、有効ターゲットの存在又は不存在を示す。GOが1ならば、22ビットフィールドは、パケットのルーティング段階及び表への挿入及び表からの削除の段階の両方において、アドレス探索にて考慮されるべき次行の指標(フィールドNEXT_ROW)を含む。GOが0ならば、フラグTARGETが0のときに22ビットフィールドは有効情報を含まず、逆に、それは検索の結果として出力にて与えられるべき有効ターゲットの指標と解釈され得る。この場合には、22ビットフィールドは、以下のサブフィールドから構成される。

【0018】VALUE: ターゲットの実際の識別(identity)を符号化する9ビット。

MASK: サブフィールドVALUEに記憶されたターゲットに関連するマスクを符号化する4ビット。kが探索ステップならば、MASK値は、0~k-1の範囲で変わり得る。k-1より小さい値は、長さがkの倍数ではないプレフィックスを示す。

T_BAK: 所謂「プロパー・ターゲット(proper ta

rget)」(又は「バックアップ・ターゲット(backup target)」)を符号化する9ビット。この用語は、「カバー」され得るターゲット、すなわち、探索ステップの倍数でない長さを有し且つ1組のより長いプレフィックスにより隠す(カバーする)ことができるプレフィックスを示す。後に説明するように、このより短いターゲットの存在に関する情報は、メモリの対応行のプレフィックスを削除した後に探索が正しくない結果を与えるのを避けるために、記憶されるべきである。情報T_BAKが記憶されているセルは、後に説明する簡単な数式により求められる。カバーされ得るプレフィックスが存在しない場合には、事前設定された値NULLが、T_BAKに割り当てられる。例えば、表が512(すなわち 2^9)ターゲットの最大にアクセスできると仮定すると、値NULLは、511、すなわち $2^9 - 1$ となる。情報T_BAKを記憶するために、ターゲットのフィールドVALUE, MASKが、セルの情報フィールドの長さより短い全体長さを有し、さもなければ使用されずに残されるセルの一部が利用されることを留意すべきである。このような特徴を可能にするべくサブフィールドVALUE, MASKの大きさを選ぶことは、構造の一般性を損なわない。

【0019】GO=1及びTARGET=1の場合は、ターゲットへの到達、同時に探索操作を継続する必要性に対応する。上述のように、セルは調べられる行の指標と到達レベルに関連の最長ターゲットを同時に含むべきでないので、第2の情報を延ばす必要がある。ターゲット情報は、フィールドNEXT_ROWにより示された行内のセル全てに共通であるので、それを「理想的に」行自身の頂部に記憶することができる。2のべきである幾つかの列を保持するために、本発明によりAUXで示されたRセルの補助ベクトルが、メモリM1に接続される。ベクトル内の各セルは、メモリM1の行に接続され、M1内のセルと同じ構造を有する。このようなベクトルは、各行に対し、調べられるべき次行の値をこのセルに記憶する必要性故に前のレベルのセルから受け継いだターゲット情報を含む。以下、AUX[NEXT_ROW(現在のセル)]なる記法は、セルに対応して見出されたターゲット情報が、セル自身を用いて到達すべき行内の補助ベクトルに記憶されることを示すのに使用される。

【0020】フラグGOとTARGETのその他全ての組み合わせに対し、セルAUX[NEXT_ROW(現在のセル)]は空である。まとめると、2つのフラグの意味は以下の通りである。

GO	TARGET	22ビットフィールド
1	0	次行を調べる
1	1	次行を調べる; AUX[NEXT_ROW(現在のセル)]は、アドレスに対する有効ターゲットを含

0 1 む
サブフィールドVALUE内のアドレスに対して最
も明確な有効ターゲット（探索終了）
0 0 空セル（探索終了）

【0021】情報T_BAKに戻ると、この情報が記憶されるべきメモリ内の位置は、探索ステップkの場合、カバーされ得るプレフィックスの数は、
【数1】

$$\sum_{j=1}^{k-1} 2^j = 2^k - 2$$

であることに基づく。従って、行内のセル数が正確に 2^k であることを利用することにより、カバーされ得るプレフィックスであって同じ行に属するものをメモリ行に記憶することができる。kの倍数の長さを有するプレフィックスは、自身が行内に記憶可能な最長プレフィックスである他のより長いプレフィックスによりカバーされ得ない。実際、kビットのアドレス部分、kより小さい部分のマスク、及び行内のセルの各々の対間の一対一対応を確立する機能が存在する。「インデックス」がアドレス部分及びその部分に対応するマスクMASK（kより小）であるならば、カバーされ得るターゲットが列内に記憶されるセルは、次式により同定される位置「オフセット」を有する。

オフセット＝インデックス－1＋ $2^{(k-MASK-1)}$

「オフセット」情報の使用は、挿入及び削除操作から生じる。

【0022】次に、本発明によるメモリ内での探索、挿入及び削除の操作を、添付されたフローチャートを参照して説明する。ここでは、以下の略記用語が適用される。

(ア) dest_add , dest_route — それぞれ宛先アドレス又は宛先出力

(イ) prou = 到達される行でのポインター（現在の行）

(ウ) ptarget = 可能なターゲットでのポインター（潜在的にカバーされ、探索自体の間に捜し出されるターゲットの探索終了まで、トレースを保つために使用される）

(エ) t_info = 出力にて与えられるべきターゲット

(オ) cell.go , cell.target , cell.value , cell.row , cell.mask , cell.t_bak = セルの同音異義のフラグ／フィールドの値（図示と表記を簡単にするため、図中、セルは、供給される情報と同数のフィールドとして理解されるように扱われている）

(カ) maskbit — 調べられているアドレス部分のマスク

【0023】1）探索（図4～6）

探索段階中、メモリは、探される宛先アドレスを入力として受信する。メモリの行0は、常に開始行であり、宛

先のkビット群（「インデックス」）が調べられたときはいずれでも、探索は行から次行に移る。すなわち、分析は、最上位kビットから開始し、なお調べられるマスクビット数がステップk以下となるまで、kビット群ごと継続する。行内の正しいセルは、調べられている群の値に対応するものである。各行にて、フラグGOが1ならば、調べられるべき次行は、現在のセルのフィールドNEXT_ROWに示される。しかしながら、次行にアクセスする前に、フラグTARGETがチェックされなければならない。後で分かるように、もしフラグTARGETが1ならば、次行が挿入段階中に現在のセルからターゲットを受け継いだことを意味する。このようなターゲットは、今までに見出された最も明確なものであり、よって、記憶されなければならない。ポインター「ptarget」の使用は、探索段階中に使用されなくなった情報を書き込むためにメモリに実際にアクセスする必要性なしに、このような記憶操作に供する。

【0024】フラグGOが0ならば、探索は終了である。いずれにしても、出力に与えられるべき情報を求めるために、フラグTARGETがチェックされる。このフラグTARGETが1ならば、現在のセルは、入力にてアドレスの最も明確なターゲットを含み、さもなければ、ターゲットは、探索段階中に既に求められており、「ptarget」により示されるAUXのセル内に記憶される。しかしながら、この第2の場合にはセルAUX内のターゲット値はNULLとなり得ること、すなわち、探された宛先の有効なプレフィックスがメモリ内に存在しないので、探索は失敗したことに留意すべきである。探索プロセスは、図5ではベクトル形式にて表されており、また図6では数値例により示されている。ここでは、簡単のため、8ビット宛先アドレスとステップk＝2が考慮されている。この図で、GとTは、フラグGOとTARGETを示す。

【0025】以下のアドレスが関連のマスクとターゲットと共に記憶され、アドレスがそれぞれ01101100と00100101である2つのメッセージのルーティングが探されることを仮定する。

アドレス	マスク	ターゲット
00100000	4	A
00100111	8	B
01101000	5	C
10000000	1	D

【0026】第1のメッセージの探索経路は、以下の通りである。

(1) 行0、セル01 → 次行＝4（GO＝1，TARGET＝0）

(2) 行4、セル10→次行=5 (GO=1, TARGET=0)

(3) 行5、セル11: このセルではGO=0及びTARGET=1; ビット対00がなお調べられるべきとしても、探索は終了しなければならない。このことは、いずれにしてもさらに特定のプレフィックスはメモリ内に存在しないことを意味する。ターゲット(第3の記憶プレフィックスに関するC)が見出され出力される。加えて、一致に到達したプレフィックスのマスクが5であるから、プレフィックス(01101)は、セル10とセル11の両方をカバーし、第1のものにおいては、ターゲット値が、フィールドT_BAKにも記憶される。T_BAK内のターゲットの記憶のため、セル10は上記関係式を満たすことが直ぐ分かる。

【0027】第2のメッセージの探索経路は、以下の通りである。

(1) 行0、セル00→次行=1

(2) 行1、セル10→この時点で探索は完了しないが(GO=1)、TARGET=1も存在する。というのは、第1の記憶アドレスのプレフィックスとターゲットAとの一致が存在するからである(第2の記憶アドレスは、たとえ第1のものと同じ初期ビットを有するとしても、マスク8を有し、よって、それとの一致は、8ビット全てを調べた後にのみ認識され得る)。従って、ベクトルAUXが使用されなければならない、見出されたターゲットのMASKとVALUEの値は、行1(行2)のセル10のフィールドNEXT_ROWにより示されたこのベクトルの行内に記憶される。

(3) 行2、セル01→次行=3

(4) 行3、セル01→セルはいかなるデータも含まず、使用されるターゲットは、AUXのものである。

【0028】2) 挿入(図7~10)

挿入の場合、メモリにロードされるエントリは、アドレス、マスク及びターゲット(アドレスに関連する出力)を含む。探索に関する操作は、メモリの初期行(行0)から開始して進み、マスク内でまだ調べられてないビット数がステップk以下になるまで、各時間を1行だけ進行させる(サブルーチンI_SEARCH_ROW)。この条件が到達された行は、新しいエントリが記憶されなければならない行を表す(サブルーチンI_MODIFY_ROW)。まず、行探索(サブルーチンI_SEARCH_ROW、図8)を考えると、行内で実行されるべき操作は、フラグGOの値に依存する。もしGO=1ならば、行は既に存在し、フィールドNEXT_ROWに示された次行は、どんな特定操作も行うことなくアクセスされなければならない。もしGO=0ならば、行は作られねばならない。作られる行は、例えば空の行のリストの最初のものであり(HEAD(row_free_list))、これはその結果更新される(UPDATE(row_free_list))。メモリに関

連する空の行リストが編成され得る方法は、技術者には周知であり、本発明の部分を構成しないので、説明は不要であろう。さらに、もしTARGET=1ならば、セルに含まれるターゲットは、新たな行に対応するアドレスにて補助ベクトルAUX内にコピーされる。いずれにしても、フラグGOは1に設定され、現在のセルが調べられる次行の指標を含むことを示す。

【0029】宛先行内で行われる操作に移ると(サブルーチンI_MODIFY_ROW、図9)、それらの操作は、調べられるビット数が実際にはkであるか又はkより小さいかに依存する。第1の場合には、実際は、挿入に係するセルは、ただ一つであり、一方、第2の場合には、操作は、セル群に係する。調べられるビット数がkならば、ターゲットは、カバーされ得ず、確かにセル内に記憶される。使用されるセルは、フラグGOの値に依存し、もしGO=0ならば、メモリの現在の行のセルとなり、もしGO=1ならば、現在のセルの内容によりアドレス指定された補助ベクトルのセルとなる。両方の場合、ターゲットとマスクはセル内に記憶され、もしGO=0ならば、ベクトルAUXのセルのフラグTARGETは、1に設定される。それ以後、現在の行のセルのフラグTARGETは、1に設定される。調べられるビット数がkより小さいならば、入力アドレスターゲットは、「インデックス」列から「オフセット」値(図中「sup」)により示された列まで現在の行における隣接したセル群全てに係する。すなわち、木に沿ってkステップ以前に到達し得るセルのインターバル全てに係する。これらのセルの各々に対し、サブルーチンI_MODIFY_CELL(図10)の操作が行われる。また、これらの操作は、GOとTARGET値により以下の様に案内される。

【0030】(ア) GOが1ならば、セルは既に行ポインタを含み、ターゲットは、フラグTARGETが0であるとき、又はTARGETが1であるがセルAUXに記憶されたマスクが入力マスク以下であるとき、AUXに記憶されなければならない。

(イ) GOが0ならば、セルは、どんな行ポインタも含まない。TARGETが1であるとき、トリ・メモリセルに記憶されたマスクが入力マスク以下ならば、入力ターゲットは、トリ・メモリセルに記憶される。

(ウ) セルが空(GO=0且つTARGET=0)ならば、フラグTARGETを1に、フィールドT_BAKをNULL値に一時的に設定する間、ターゲットが記憶される。一旦最後のセルに到達したら(図9、iからの出力「no」<sup>)、関連フラグGOがなおチェックされ、ターゲットが、GOが0又は1のどちらかによって、現在の行又は補助ベクトルのセルのフィールドT_BAKに記憶される。行の修正操作は、メモリが0に初期化されることを仮定して説明してきたことに留意すべきである。さもなければ、T_BAKの値のチェックが

必要となり得る。

【0031】3) エントリの削除(図11~15)
エントリの削除のため、要求されたエントリを含む行に到達するまで、メモリを走査しなければならない。探索又は挿入に関し、最終行は、マスクビットがk以下であることにより示される。削除とは、セル又は1組のセルからターゲットの値MASK, VALUEを削除すること、及び必要ならばそれらを取り消されるべきものでカバーされた明確性のより低いターゲットの値と置き換えることである。これらの操作を正しく行うために、調べられる行のトレースを保つことが必要であり、この目的のため、ベクトル「STEP」、及び調べられる行を含んだカウンター「ACTUAL」を使用する。ベクトル「STEP」は、メモリ行と同数のセルを有し、調べられる行のアドレス(prow)を記憶する。

【0032】一般的な削除プロセスを図11に表す。新しい行が調べられるときはいつでも、関連アドレスが、ベクトル「STEP」の適当な位置に記憶される。全ての到達セルに対し、フラグGOもまたチェックされる。このフラグGOが0ならば、削除操作は、実際には記憶されてない宛先アドレスに関係することを意味し、これは明らかに操作を終了させる。一旦宛先行に到達したならば、代用ターゲットの存在が同じ行内で探され(サブルーチンD_SEARCH_T_BAK)、プレフィックスが挿入された行は、なんらか見出された代用ターゲットを用いることにより修正され(サブルーチンD_MODIFY_ROW)、後方チェックが、空となった行に対して行われ(サブルーチンD_BACKTRACE)、それらを除去する。サブルーチンD_SEARCH_T_BAK(図12)では、削除されるエントリのプレフィックスを識別するセルが調べられる。このプレフィックスは、メモリ構成ゆえに、特定の行に記憶され得る。すなわち、可能なより短いプレフィックスをT_BAKに有し得るセルが調べられる。この探索は、前の行から現在の行に導かれた探索木の一部に対応するセルを調べることを要する。探索は、(GOの値によってメモリのセル又は補助ベクトルのセルにおいて)有効ターゲットが見つけれられるまで、又はターゲットを見出すことなく全てのセルが調べられるまで、続けられる。特定セルのフィールドT_BAKがNULL値を有しないならば、それぞれのマスクに関連する代用ターゲットが見つけれられる。ターゲットの検索は、フラグ「found」を1に設定することにより、伝えられる。

【0033】挿入の場合の行修正(サブルーチンD_MODIFY_ROW、図13)では、マスクがただ1つのセル又はセル群を識別するか否かを確証するために、ビット群が正確にkビットを有するかkビットより少ないかをチェックしなければならない。第1の場合には、エントリがベクトルAUX内か実際のメモリセル内のどちらで探索されるのかを、値GOにより区別する必要がある。

ある。代用ターゲットは、前のサブルーチンで見出されたならば(フラグ「found」が1)、これは削除されるべきターゲットを置き換え、操作は終了する。一旦削除操作が行われると、代用ターゲットが見つからなかったならば、行が空になっていないことをさらに確かめる必要がある。maskbit < kの場合、削除されるエントリは、メモリ又はベクトルAUXの対応セル内のアドレス「offset」と共に、セルのT_BAKフィールド内に記憶される。「sup」インターバルに対応するセル全て(サブルーチンD_MODIFY_CELL)が、チェックされる。代用ターゲットが見つ出されたならば、セルのVALUE及びMASKフィールドが修正される。ターゲットが見つ出されなければ、フラグGOとTARGETの両方とも0に設定され、セルが空であることを示し、加えて、T_BAKフィールドがNULLに設定される。

【0034】サブルーチンD_MODIFY_ROWの実行中は、リクエストは非存在要素に関係しないこともチェックすることに留意すべきである(これは、maskbit=kの場合、TARGET=0又はMASK≠k-1又はVALUE≠t_infoにより示され、maskbit<kの場合、T_BAKフィールド内のターゲット値の不存在により示される)。これらの場合には、明らかに探索はすぐに終了する。最後に、空の行を除去するために(サブルーチンD_BACKTRACE、図15)、探索において調べられる行全てが、後方トレースされ、それらの各々に対し、全てのセルのフラグGOとTARGETがチェックされる。行内の全てのセルのフラグが0のとき、行は空であり、セル「father」に戻る必要がある。このセル「father」から空の行への移行が生じる。この目的のため、カウンターACTUALとベクトルSTEPが用いられる。このセル「father」に対し、TARGETフラグが1ならば、AUXの対応セルのデータは、このようなセルに転送されなければならない。これが一旦行われると、空の行の識別が、空行リストに加わり(prow = TAIL(row_free_list), UPDATE(row_free_list))、次行、すなわちセル「father」を含んだ行が調べられる。少なくともセルが空でない行に一旦到達すると、操作は終了する。

【0035】上記記載したメモリを実現するハードウェアについて考察するのが有益と思われる。アーキテクチャーは、本質的に状態機械(図1のコントローラCTの一部を作る)により構成される。この状態機械は、外部メモリ(図1中のM1)に記憶されたデータ構造へのアクセスシーケンスを制御する。この外部メモリは、ダイナミックRAMメモリとするのが有利である。アーキテクチャー全体は、3進CAMメモリとして動作する。メモリの初期化と更新は、簡単のためコントローラCTに組み込まれた別の状態機械又は従来のマイクロプロセッサのどちらかにより制御され得る。アーキテク

ャーは極度に柔軟である。というのは、任意長のストリングが受信され得る一方、出力はインデックス（又は圧縮された識別子）であり、その意味は自由であり、アプリケーションの内容に依存するからである。インターネット網を考える場合、出力は、通常は表M2へのポインターであり、該表M2が、パケットを入力識別子として与えられた宛先アドレスに発送するのに必要とされる次のホップデータ（レベル2アドレス）を含む。外部データ構造は、行の全てのセルに共通なターゲット情報を含む補助セルと共に、プログラム可能な数の行により構成される。これらの行の各々は、2^k の同じ隣接セルを含む。行アドレスとセルアドレスを並べて置くことによりセルアドレスを得るために、行の補助ターゲットセルは、同じデータ構造内に、例えば実際のデータを含む最終行の後に別々に記憶される。メモリの大きさは、プレフィックスの分布と探索ステップに依存する。指示タイトルとして、約40,000エントリを有する発送表では、 $k=4$ のとき、約14,000行（約0.7Mbytesのメモリに対応）を要し、 $k=8$ のとき、約3,000行（全体量約2.4Mbytesのメモリの場合）を要する。従って、大きさは完全に制御可能である。

【0036】

【発明の効果】上記記載したことから、本発明による利点は以下の通り明らかである。

（1）セルの内容を特徴付けるための2つのフラグを採用すること、及び探索段階の終了前に見出されたターゲットが記憶されるという例外ケースに対し補助ベクトルを使用することにより、メモリが非常にコンパクトになり、場合によっては起こりうる情報フィールドのダブりに固有の浪費が避けられる。

（2）ハードウェア構成において補助ベクトルは、単に実際のトリ・メモリのセルの追加ストリングを構成するということを考慮すると、最長プレフィックス一致が起これないときは、同じ装置が固定長マスクに対しても使用できる。すなわち、唯一の差は、この場合には補助ベクトルは決してアドレス指定されず、この場合の構造は、通常の2進CAMとして働き、1セル当たり1ビットを浪費することである。

（3）フィールドT_{BACK}により、場合によってはより短いプレフィックスを有するターゲットの存在が、挿入及び削除の操作において扱われ、これらは、予め限定された所定数のセルを調べることをのみを要求する（ k が探索ステップならば、 $k-2$ ）。

（4）探索フラグの明確な終了を用いることにより、構造は、アドレス又はマスクの特定の長さには結びつけられず、好適用途に関係するものに関し、128ビットアドレスが採用されるときにも利用可能となる。

（5）最後に、挿入及び削除のメカニズムは、特に簡単であり、局所化されている。すなわち、メモリ又はその一部の完全な書き込みを要求しない。

【0037】記載してきたことは単に非限定的な例として与えられていること、及び本発明の範囲を逸脱することなく変更や修正が可能であることは自明である。特に、たとえインターネット網が参照されてきたとしても、パケットのルーティングが固定又は可変長のプレフィックス一致、例えばATM接続又はMACアドレス（MAC=Medium Access Control、これは標準化通信プロトコルにおける層である）の識別に基づいている如何なる通信プロトコルにも本発明を適用できる。一般に、本発明は、通信システムのノードだけでなく、情報探索又は検索がプレフィックス（例えば、通信網に属していないデータベースから検索できる情報）を用いて実行される全ての場合に対して、たとえこれらの用途において通信網により必要とされる速度要求が一般に存在しないとしても、適用できる。

【図面の簡単な説明】

【図1】インターネット網へのルーティングノードの基本図である。

【図2】本発明によるメモリの編成を示す図である。

【図3】メモリセル内のデータ編成を表す。

【図4】図2のメモリ内での探索操作に関する図である。

【図5】図2のメモリ内での探索操作に関する図である。

【図6】探索の数値例である。

【図7】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【図8】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【図9】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【図10】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【図11】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【図12】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【図13】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【図14】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【図15】本発明によるメモリ内でのデータ挿入及び削除に関するフローチャートである。

【符合の説明】

IFa... IFx 入力インターフェース

IFb... IFy 出力インターフェース

SW スイッチング装置

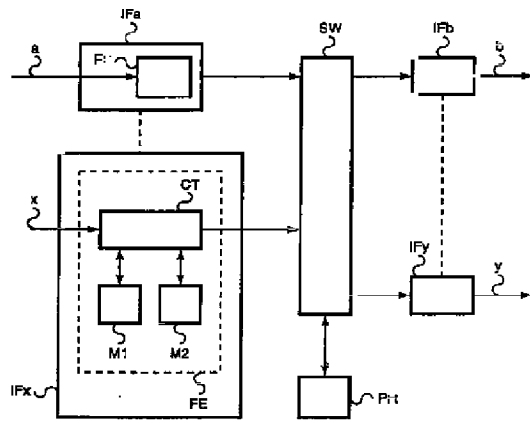
PR 制御装置

FE ルーティング探索装置

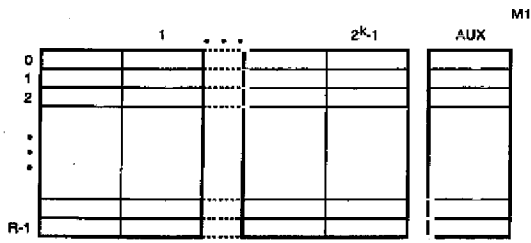
M1, M2 メモリ

CT 処理制御装置 (コントローラ)

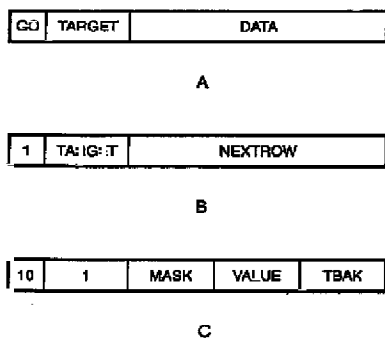
【 図 1 】



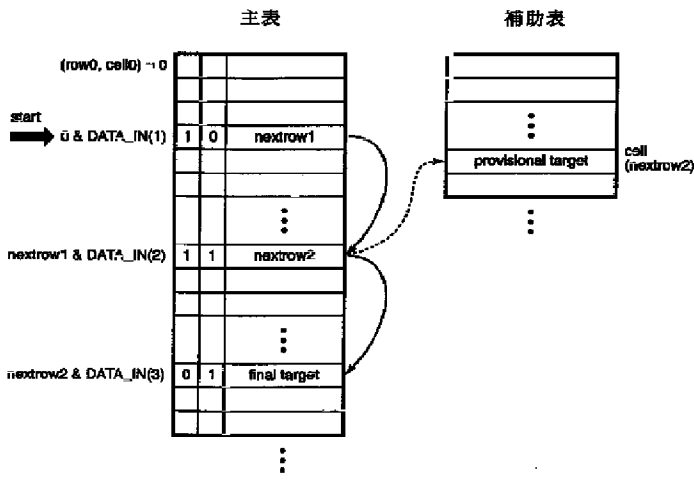
【 図 2 】



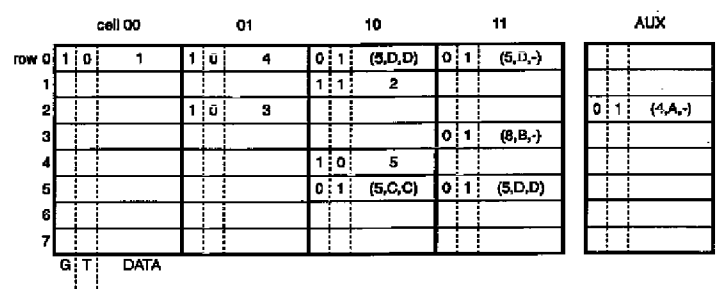
【 図 3 】



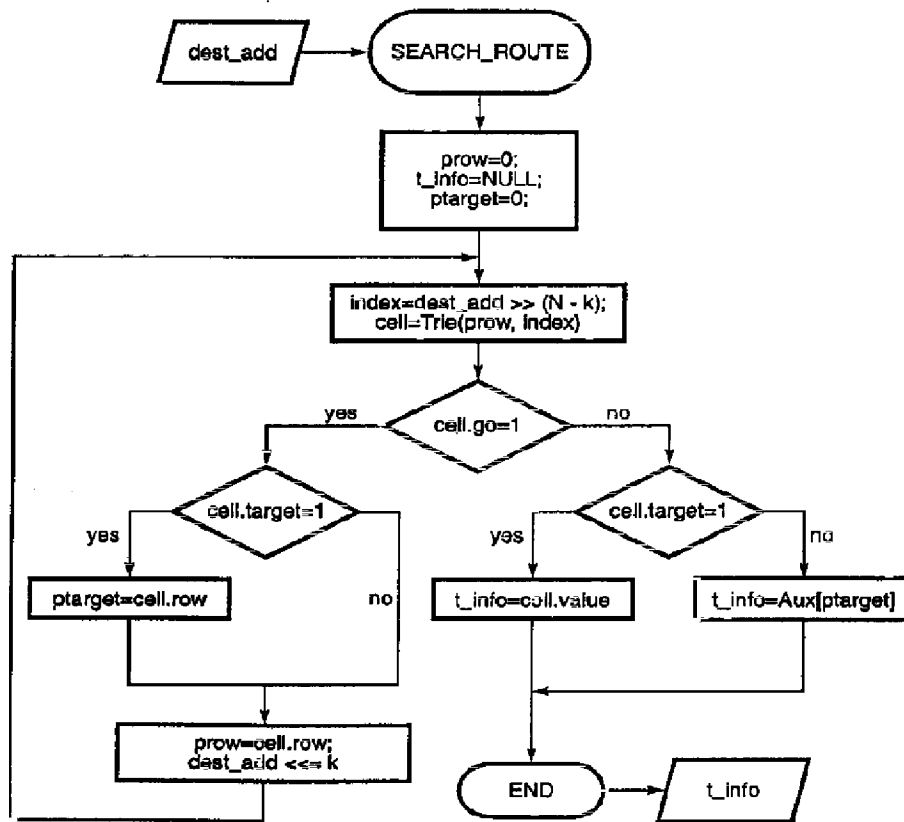
【 図 5 】



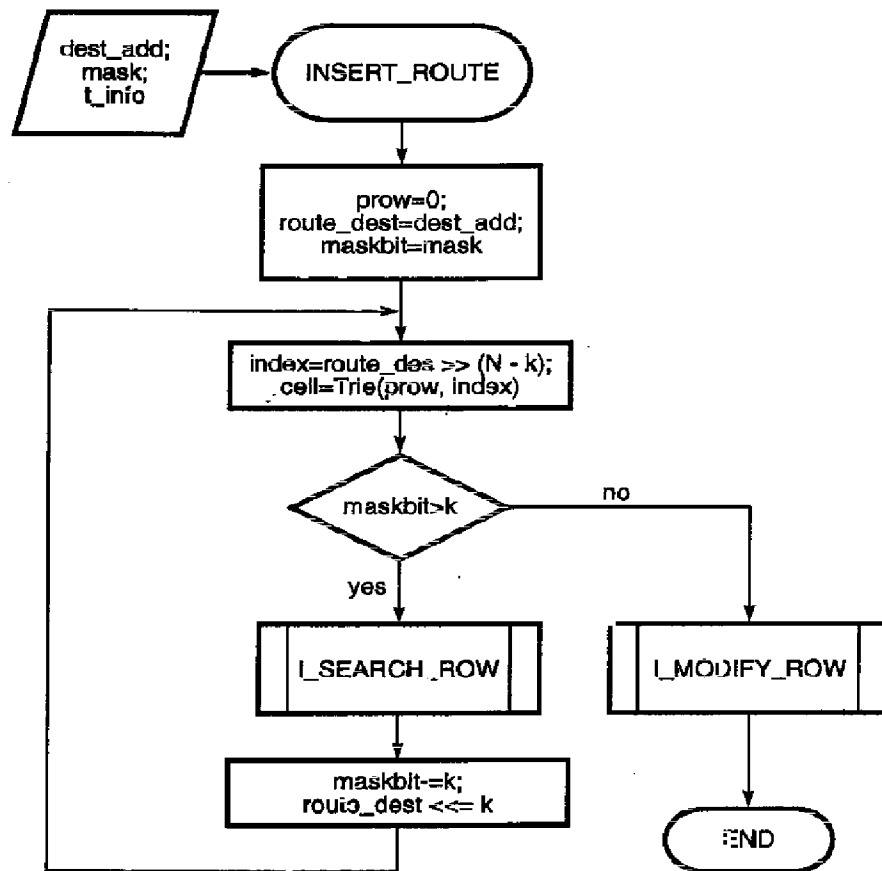
【 図 6 】



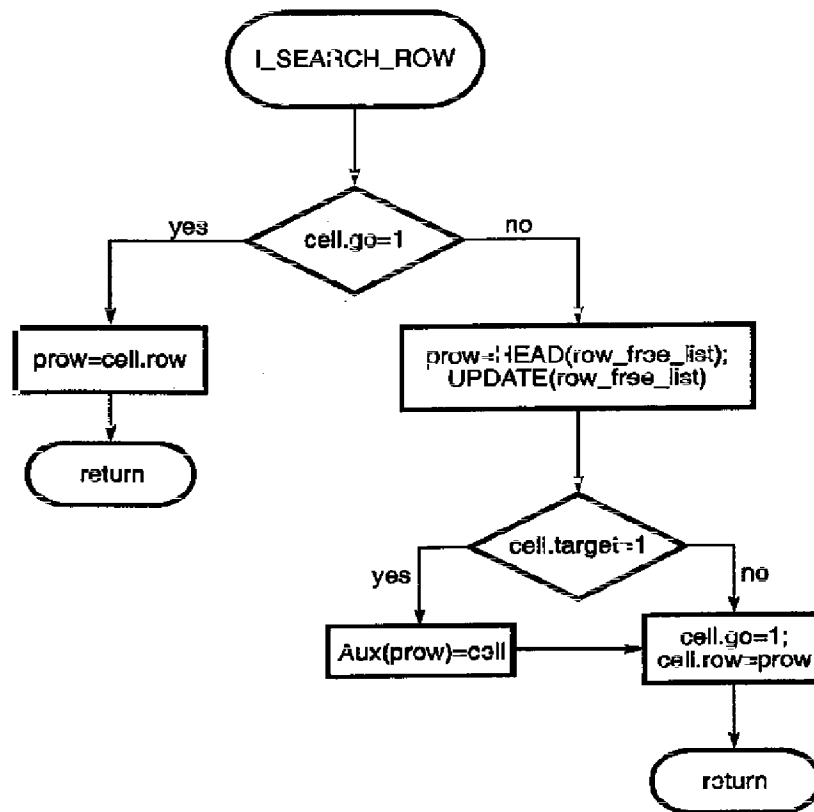
【図4】



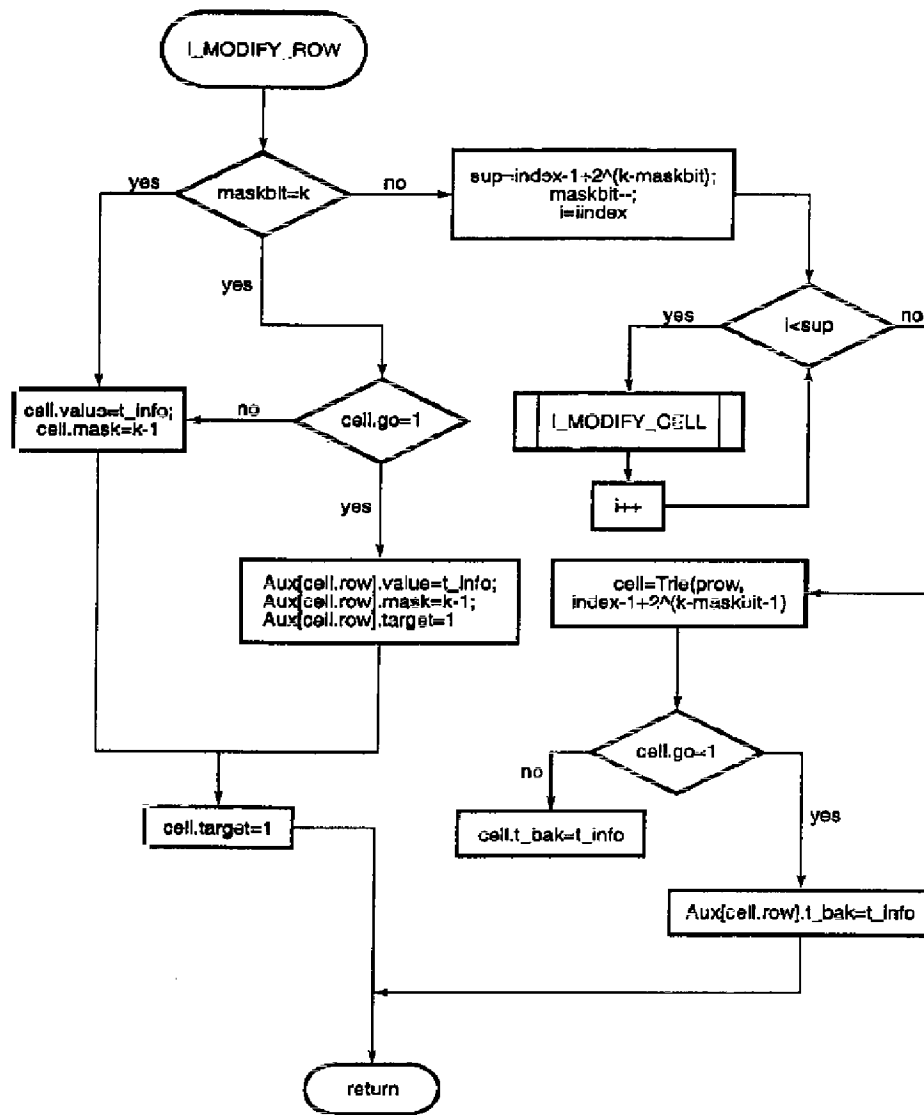
【図7】



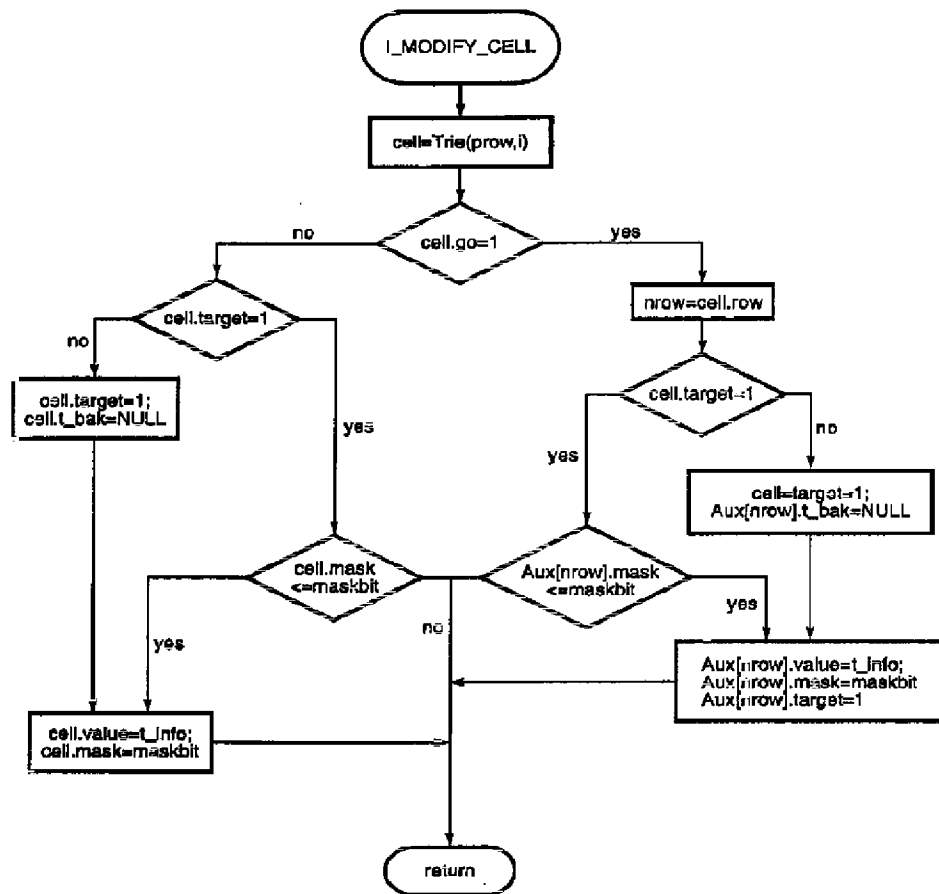
【圖8】



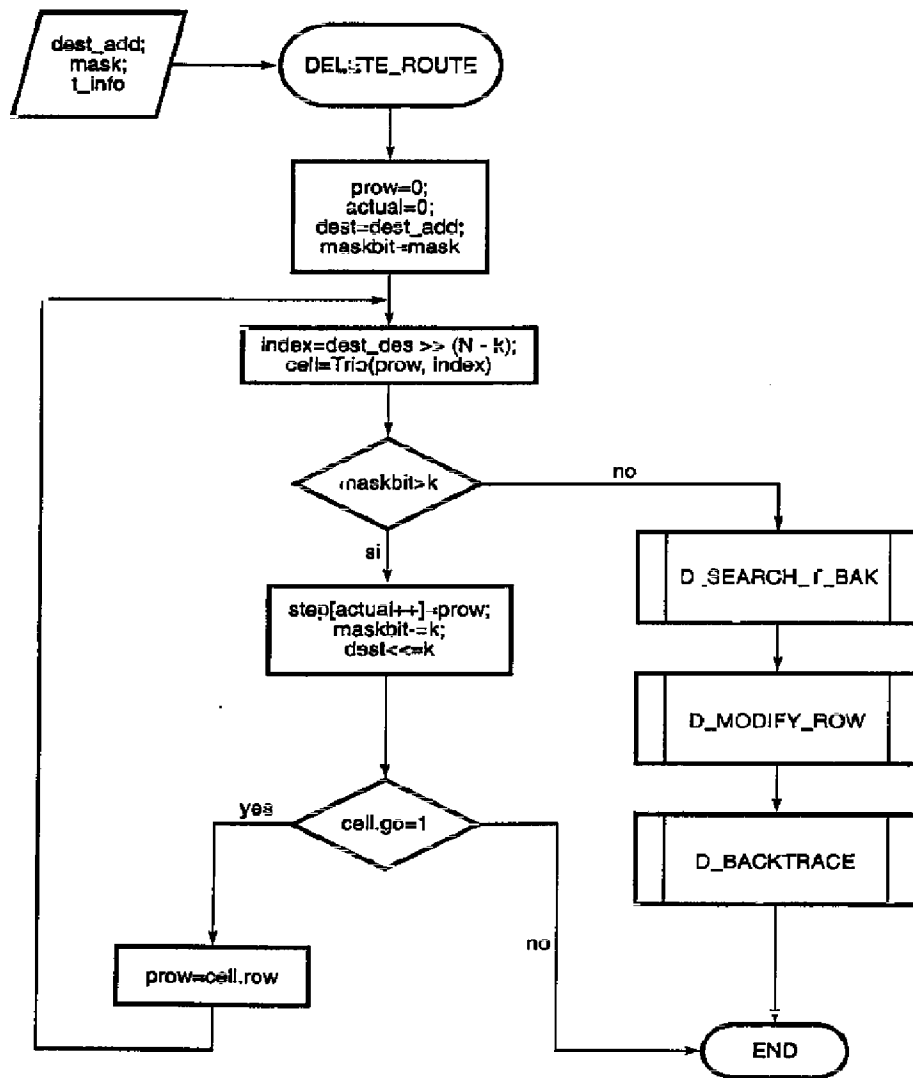
【図9】



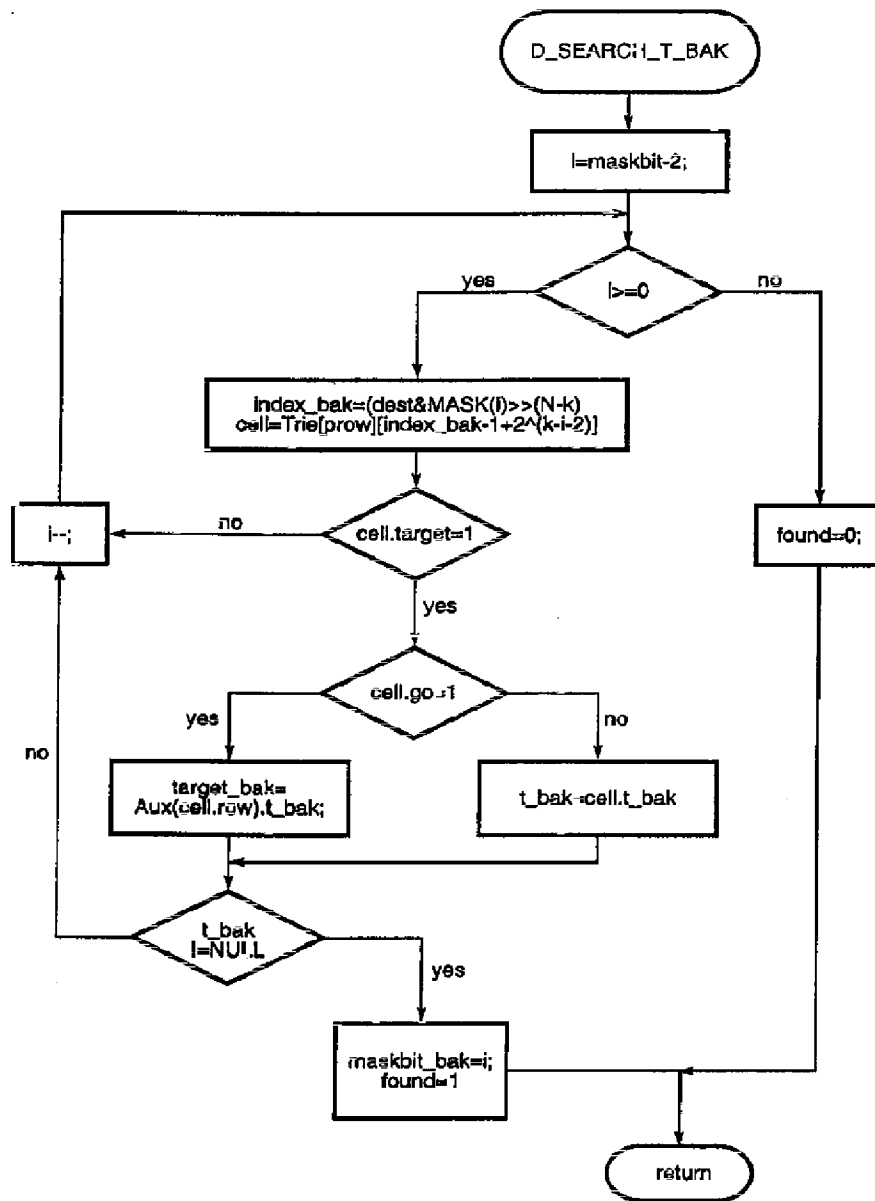
【 図 10 】



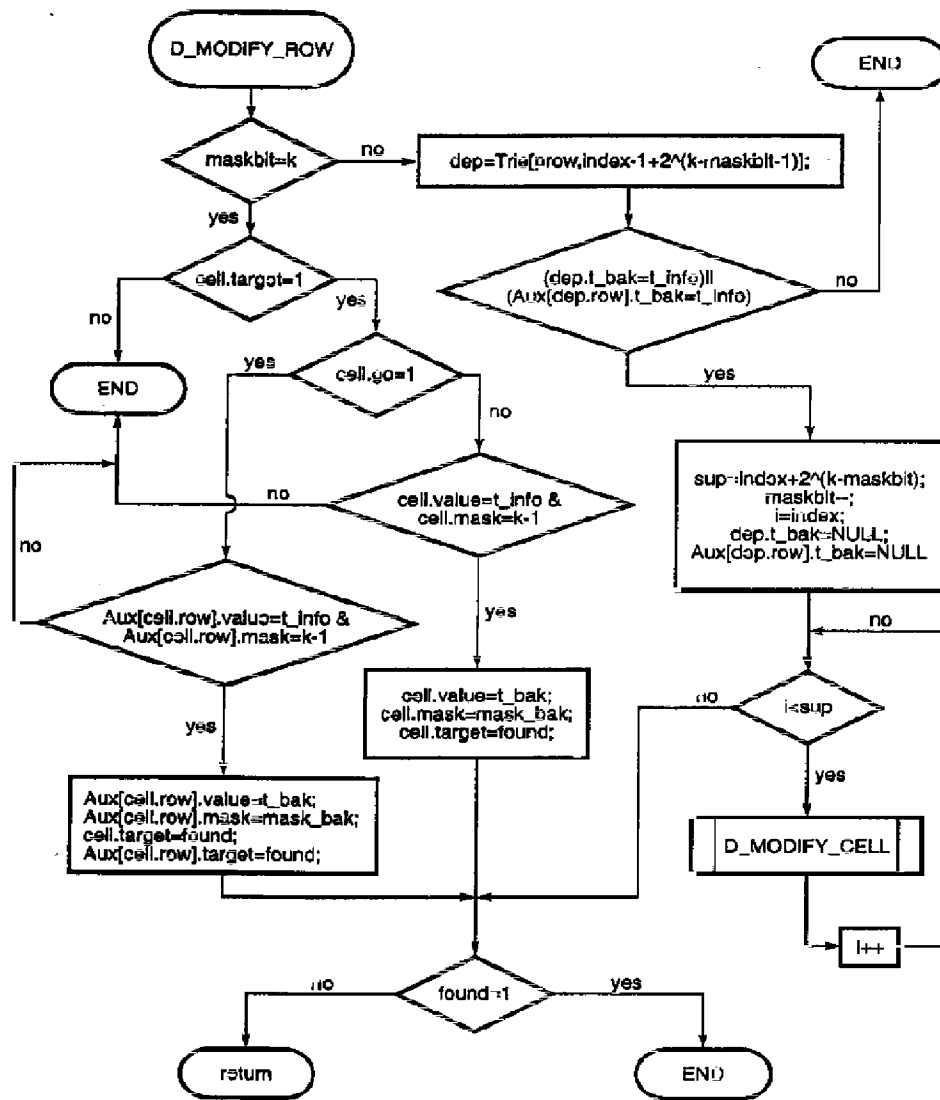
【 図 11 】



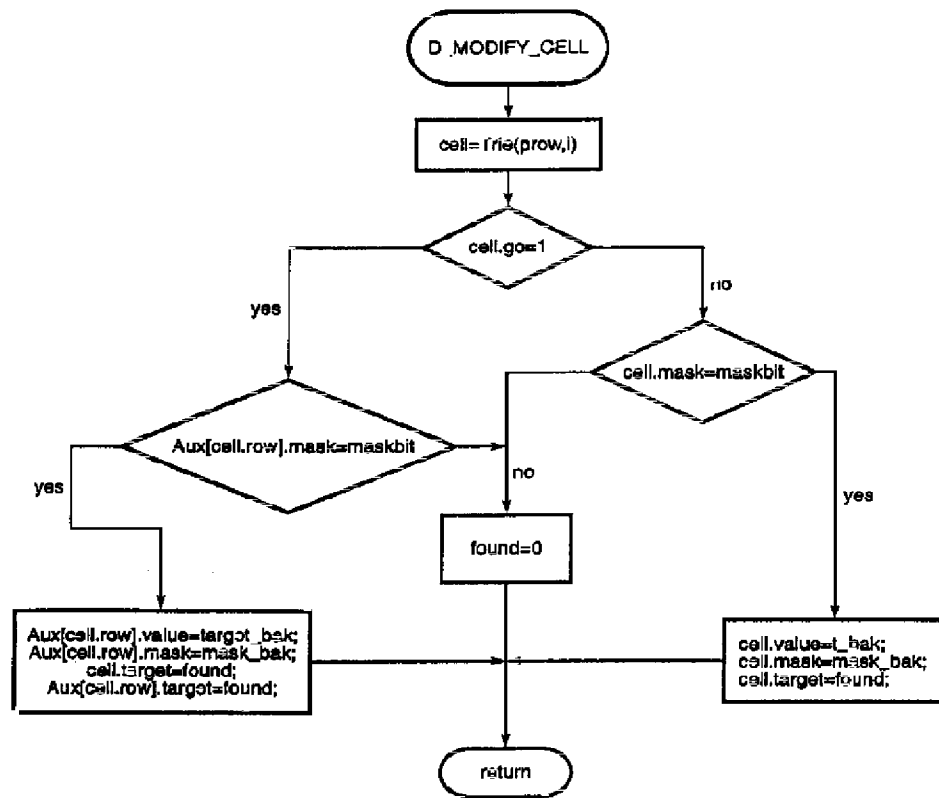
【図12】



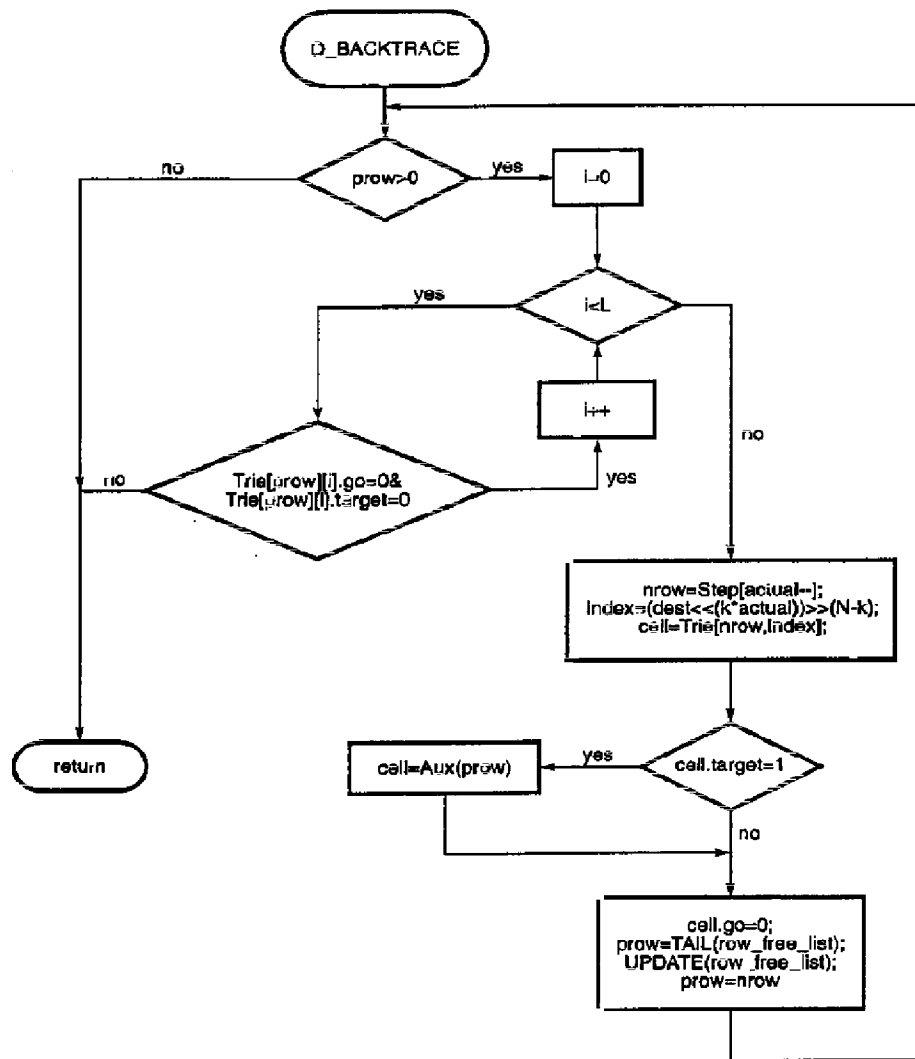
【 図 13 】



【図14】



【図15】



フロントページの続き

(72)発明者 エンリカ・フィリツピ
 イタリア国 10128 トリノ、シー・エツ
 セオ・モンテヴェツキオ 49

(72)発明者 ヴィヴィアナ・インノセンティ
 イタリア国 10040 リヴァルタ(トリ
 ノ)、ヴィア・アレンデ 8